Hardware Reference Manual

VICTOR Publications
380 El Pueblo Road
Scotts Valley, CA 95066
(408) 438-6680

Second VICTOR printing April, 1983.

ISBN 0-88182-065-2

Printed in U.S.A.

CONTENTS

FIGURES

TABLES

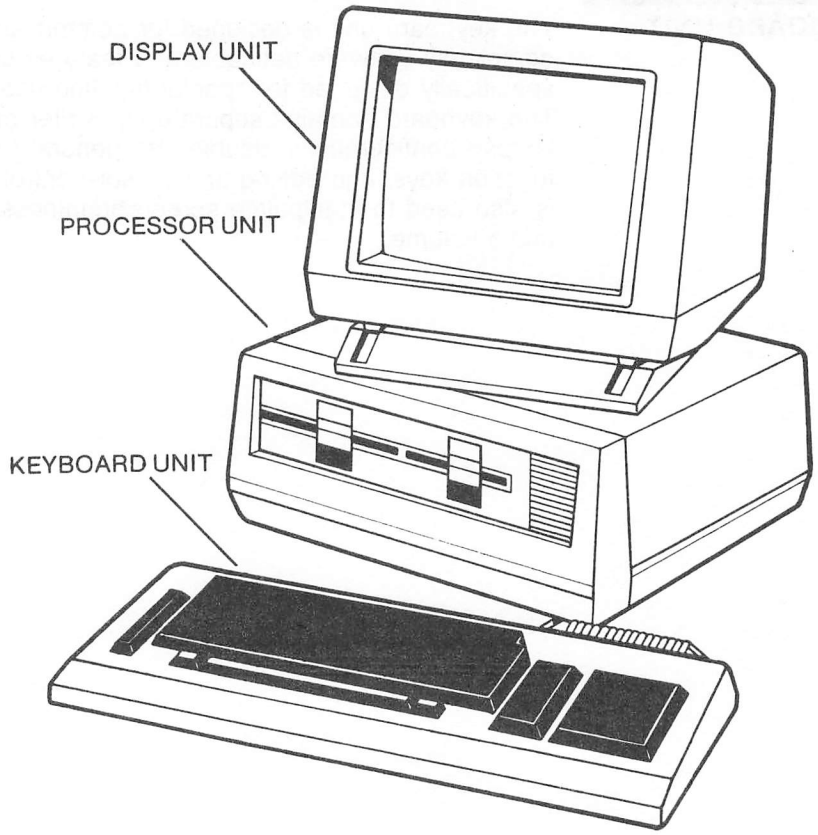
1

SYSTEM DESCRIPTION

1. SYSTEM DESCRIPTION

The system is designed for maximum operator comfort and comfort and ease of use. The system is composed of three modules, and occupies the desk space normally needed for an office typewriter. Its modules are: the processor unit, the display unit, and the keyboard unit. Coiled cables interconnect these stand-alone modules, allowing easy positioning and mobility. A standard configuration is shown in Figure 1.

Figure 1: Typical Arrangement of Main Units



The system can be connected to a wide variety of peripherals and accommodates local and long distance communications. Standard interfaces include a parallel port (Centronics or IEEE-488), programmable RS-232(V-24) channels, an internal control port, and an audio controller for digitized voice and tone output.

PROCESSOR UNIT

The processor unit physically supports the display unit, as shown in Figure 1. The main logic, disk drives, and power supply are housed in the processor unit. The two integral single-sided 5 1/4-inch floppy disk drives store up to 1.2 megabytes of information. The system incorporates a minimum 128K bytes of random access memory (RAM), expandable to 512K bytes.

DISPLAY UNIT

The display unit swivels and tilts to permit optimum adjustment of the viewing angle, and the unit incorporates a 12-inch antiglare screen to prevent eye strain. The display is 25 lines; each line has 80 characters. Characters are formed in a 10-x-16 font cell, providing a high resolution display. A bit-mapped graphics mode with 800-x-400-dot matrix screen resolution is available under software control. Software also controls the overall screen brightness, character contrast, and audio volume.

KEYBOARD UNIT

The keyboard unit is designed for comfort and ease of operation. It is completely software definable and features several keys that are specifically designed for special-function use in application programs. The keyboard contains separate typewriter and numeric/calculator keypad configurations, double-size general-function keys, special-function keys, and editing and cursor-control keys. A cluster of keys is also used to manipulate screen brightness, character contrast, and audio volume.

2

PROCESSOR UNIT

2. PROCESSOR UNIT

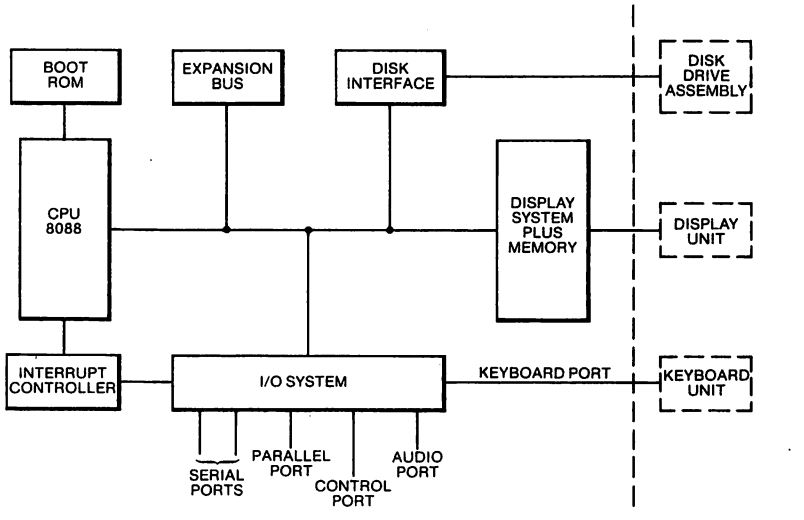
The heart of the processor unit is the Intel 8088 microprocessor. This processor is a version of the Intel 16-bit 8086 processor that contains an 8-bit bus interface. The 8088 is software-compatible with the 8086, and thus supports 16-bit operations, including multiply and divide. The processor has a 20-bit physical address space, providing 1 megabyte of addressable memory I/O.

As indicated earlier, the processor unit is the module that physically supports the display unit. It contains three basic assemblies: the main logic board, the disk drive assembly, and the power supply.

MAIN LOGIC BOARD

As shown in Figure 2, the main logic board is comprised of the central processing unit (CPU) section, the input/output (I/O) section, the display section, the disk interface section, and the expansion bus.

Figure 2: Main Logic Block Diagram



8088 CENTRAL PROCESSING UNIT (CPU)

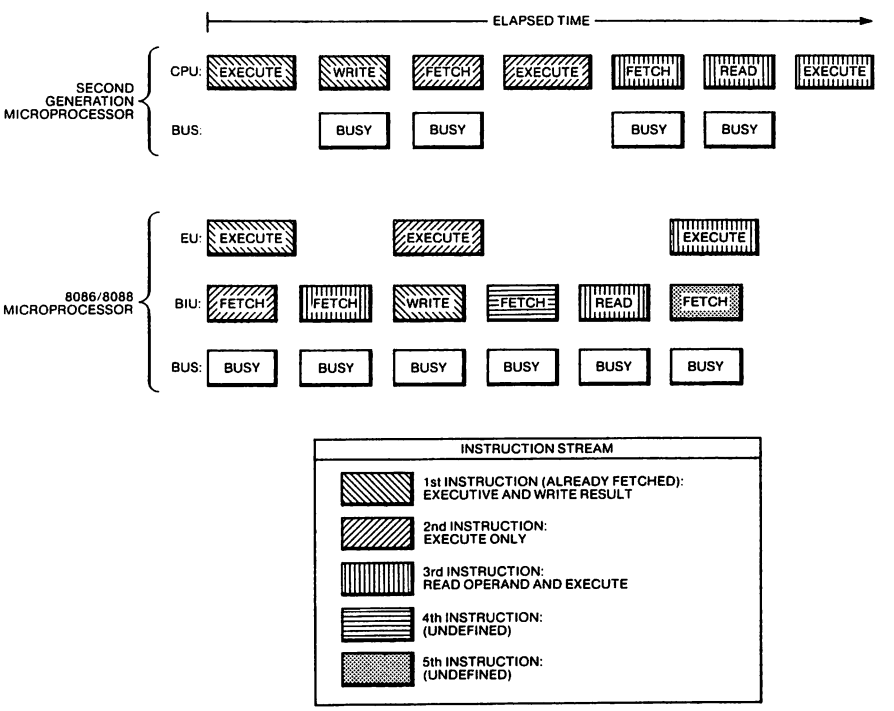
Microprocessors execute programs by cycling through the following four steps:

1. Fetch the next instruction from memory.
2. Read an operand (if required by the instruction).
3. Execute the instruction.
4. Write the result (if required by the instruction).

These steps have historically been performed in a series or with a single bus cycle fetch overlap. The architecture of the 8088 CPU allocates the same steps to two separate processing units within the CPU. The execution unit (EU) executes instructions. The bus interface unit (BIU) fetches instructions, reads operands, and writes results.

The two units operate independently of each other, thus allowing overlap of instruction-fetch activity and instruction-execution activity. The time required to fetch instructions "disappears" because it no longer impacts instruction execution time; the next instruction to be executed by the EU has always already been fetched by the BIU. Figure 3 provides an example which illustrates this overlap and compares it to traditional microprocessor operation. In the example, overlapping reduces the elapsed time required to execute three instructions, and, during that execution time, allows two additional instructions to be fetched.

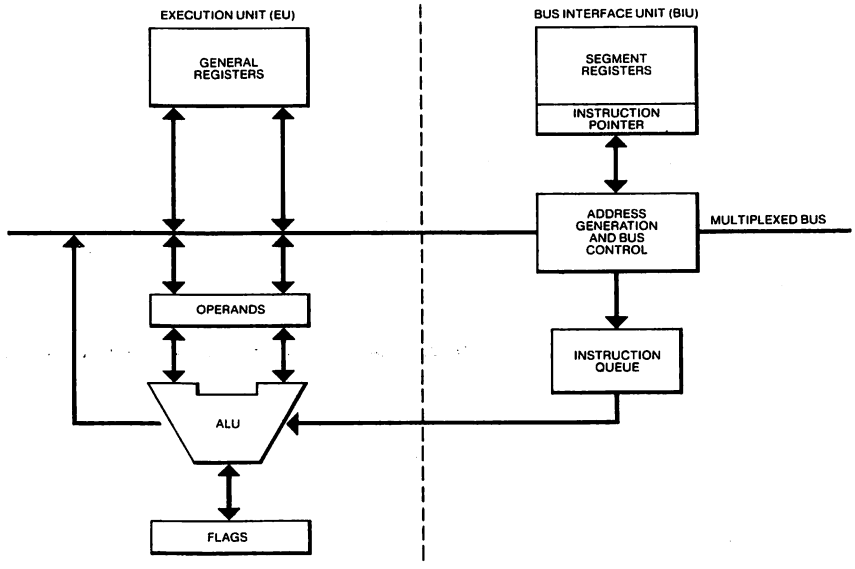
Figure 3: Overlapped Instruction Fetch and Execution



Execution Unit

All registers and data paths in the EU are 16 bits wide, providing for fast internal transfers. CPU status and control flags are maintained in the EU by a 16-bit arithmetic/logic unit (ALU) that manipulates the general registers and the instruction operands (Figure 4).

Figure 4: Execution and Bus Interface Units



The EU is not connected to the outside world via the system bus. It obtains instructions from a queue maintained by the BIU. When an instruction requires access to memory or to a peripheral device, the EU sends a request to the BIU to store or obtain the data. The BIU performs an address relocation that gives the EU access to a full megabyte of memory space.

Bus Interface Unit

The BIU performs all bus operations for the EU. Upon demand from the EU, the BIU transfers data between the CPU and the memory or an I/O device.

While the EU is executing instructions, the BIU fetches instructions from memory. The instructions are stored in an internal RAM array called the instruction stream queue. The 8088 instruction queue holds up to four bytes of the instruction stream. The queue size is sufficient to allow the BIU to keep the EU supplied with fetched instructions without monopolizing the system bus. The BIU fetches another instruction byte whenever: (1) one byte in the queue is empty and (2) there is no active request for bus access (Figure 3).

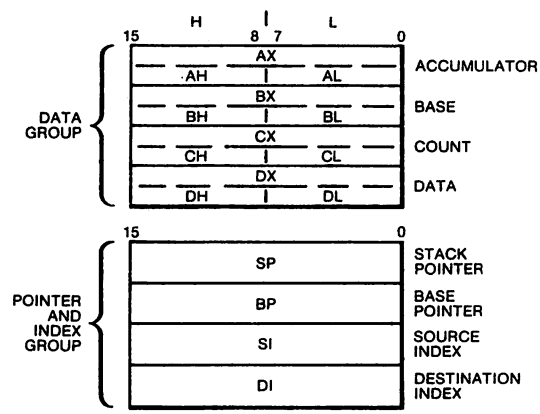
The instruction queue usually contains at least one byte of the instruction stream; the EU does not have to wait for instructions to be fetched. The instructions in the queue are those stored in the memory locations immediately adjacent to and higher than the instruction currently being executed. That is, the queue contains the next logical instructions, as long as execution proceeds serially. If the EU executes an instruction that transfers control to another location, the BIU resets the queue, fetches the instruction from the new address, passes it immediately to the EU, and then begins refilling the queue from the new location.

The BIU suspends instruction fetching whenever the EU requests a memory or I/O read or write. A fetch already in progress is completed before the EU's bus request is executed.

General Registers

The 8088 has eight 16-bit general registers (Figure 5). The general registers are divided into two sets of four registers: the data registers called the H&L group (H&L stands for "high and low"), and the pointer and index registers which are called the P&I group.

Figure 5: General Registers



The data registers are unique in that their upper (high) and lower halves are separately addressable. Each data register can be used interchangeably as a 16-bit register or as two 8-bit registers. However, the CPU registers are always accessed as 16-bit units. Data registers can be used without constraint in most arithmetic and logic operations. Certain instructions use specified registers implicitly (see Table 1), allowing compact, powerful encoding.

Table 1: Implicit Use of General Registers

REGISTER	OPERATIONS
AX	Word multiply, word divide, word I/O
AL	Byte multiply, byte divide, byte I/O, translate, decimal arithmetic
AH	Byte multiply, byte divide
BX	Translate
CX	String operations, loops
CL	Variable shift and rotate
DX	Word multiply, word divide, indirect I/O
SP	Stack operations
SI	String operations
DI	String operations

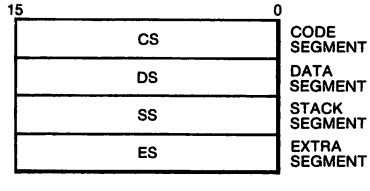
The pointer and index registers can also participate in most arithmetic and logic operations. All eight general registers fit the definition of "accumulator," as used with first and second generation microprocessors. The P&I registers (except for the BP register) are also used implicitly in some instructions, as shown in Table 1.

Segment Registers

One megabyte of memory space is divided into logical segments of up to 64K bytes each. The CPU has direct access to four segments at a time. The starting location (the base address) of each segment, is contained in the segment registers (see Figure 6). The CS register points to the current code segment; instructions are fetched from this segment. The SS register points to the current stack segment; stack operations are performed on locations in this segment. The DS register points to the current data segment and generally contains program variables. The ES register points to the current extra.

The segment registers can be accessed by programs and manipulated with several instructions.

Figure 6: Segment Registers



Instruction Pointer

The 16-bit instruction pointer (IP) is similiar to the program counter (PC) in the 8080/8085 CPUs. The IP points to the next instruction. It is updated by the BIU so that it contains the offset (distance in bytes) of the next instruction from the beginning of the current code segment. During normal execution, the IP contains the offset of the next instruction to be fetched by the BIU. Whenever the IP is saved on the stack, it is automatically adjusted to point to the next instruction to be executed. Programs do not have direct access to the IP; however, instructions cause the IP to change and to be saved on and restored from the stack.

Flags

The 8088 has six 1-bit status flags that the EU posts (Figure 7). The flags reflect specified properties of the result of an arithmetic or logic operation. Different instructions affect the status flags differently. Another group of instructions is available that allows a program to alter its execution, depending on the result of a prior operation. This result is indicated by the state of these flags. Examples of conditions reflected by the flags are described below:

- The auxiliary carry flag (AF) is set when a carry out of the low nibble into the high nibble or a borrow from the high nibble into the low nibble of an 8-bit quantity (low-order byte of a 16-bit quantity) has occurred. This flag is used by decimal arithmetic instructions.

- The carry flag (CF) is set when a carry out of, or a borrow into, the high-order bit of the result (8- or 16-bit) has occurred. This flag is used by instructions that use the CF to add and subtract multibyte numbers. Rotate instructions also isolate a bit in memory or in a register by placing it in the CF.

- The overflow flag (OF) is set when an arithmetic overflow has occurred; that is, a significant digit has been lost (i.e., the size of the result exceeded the capacity of its destination location). An interrupt on overflow instruction is available to generate an interrupt in an arithmetic overflow.

- The sign flag (SF) is set when a result's high-order bit is a 1. Negative binary numbers are represented in the 8088 in standard two's complement notation. SF indicates the sign of the result (0=positive, 1=negative).

- The parity flag (PF) is set when the result has even parity (an even number of 1-bits).

- The zero flag (ZF) is set when the result of the operation is 0.

Three additional control flags (Figure 7) can be set and cleared by programs to alter processor operations:

- Setting the direction flag (DF) causes string instructions to auto-decrement (to process strings from high addresses to low maskable) interrupt requests. Clearing IF disables these interrupts. IF has no affect on nonmaskable interrupts generated externally or internally.